Getting Started

Your first login, the console and the shell, and where to go next · Checked against tacctl 0.2.25

Contents

1. Get on 2

1.1 The login console 2

1.2 The system shell and tacctl shell 3

1.3 The password cache 3

2. Your password 4

2.1 A required password change 4

3. What your tier allows 5

4. First commands 5

5. Conventions every command follows 6

6. Where to go next 7

This page gets a new operator, engineer or support person working with tacctl: how to get on, what your tier lets you do, the first commands to run, and the conventions every command follows.

> **Example names used on this page** Server: `tacacs.example.net` · Scope: `site-a` · Devices: `sw1` (Cisco), `rtr1` (Junos) · You: `alice`

1. Get on

1.1 The login console

Log in to the tacctl server with ssh. The **tacctl console** is the login shell of every tacctl user on the server, so your login lands there:

```
$ ssh alice@tacacs.example.net
tacctl console on tacacs — type 'help'. Devices: device list. This session is logged.
tacacs>
```

At the console prompt, type commands **without** the `tacctl` prefix, for example `device list` or `ssh sw1`. The prompt shows the server's name: its host name, unless a superuser set another with `tacctl server name`.

Key or command	What it does
`Tab`	Completes commands and names (devices, users, scopes)
`?`	Shows what can come next, and the tier a command needs. On a terminal command names, sub-commands, fixed words and options are teal and live names (users, devices) bold.
`-h` / `--help` after a command	Prints the same help as `?` at that place (`user add --help`)
`help` / `help <command>`	Lists the commands **your tier** can run / one command's usage
`man` / `man <topic>`	Opens the manual page, at a command or heading if you name one (`man device config plan`, `man tiers`)
`passwd`	Changes your own password (asks for the current one first)
`console password-cache on`	Keeps your network password for the session, where your tier is allowed it (§1.3)
`console forget`	Drops your cached password now
`exit`	Leaves the console

What the console does not do:

- It runs tacctl commands and ssh to registered devices, nothing else. `scp`, `sftp`, `rsync` and other programs are refused.
- An idle session ends after a set time (30 minutes by default).
- Every session is logged.

If the console opens with `A password change is required of your account.`, change your password first (§2.1): nothing else runs until you do.

To run a single line without opening the console:

```
ssh alice@tacacs.example.net 'device list'      # one tacctl line, no terminal needed
```

1.2 The system shell and `tacctl shell`

The console word `system-shell` starts your own system shell on the server (`exit` brings you back to the console). Superusers always have it. Read-only and operator users have it only when a superuser has opened it for their tier or for them; engineers never have it. Without it, the console answers that the system shell is not available for your tier.

In a system shell, `tacctl shell` gives you the same kind of prompt (`tacctl>`), with Tab completion and `?` help:

```
tacctl shell                    # interactive prompt; leave out the word "tacctl"
tacctl shell --password-cache   # type your network password once for this session
tacctl shell -c 'device list'   # run one line and exit
```

Each line runs through `sudo`, so it is logged and checked against your tier exactly as it would be from bash. The rest of this documentation shows the full `tacctl …` form; in the console or the shell, leave out `tacctl`.

In bash itself, `tacctl <command> --help` prints the same help as `?`, and the `?` key on a line that starts with `tacctl` prints the help of the words before the cursor and leaves the line as it is. `TACCTL_HELP_KEY` moves that help to another key, or turns it `off` (Shell and completion (repo)).

1.3 The password cache

When you run many device logins in a row (a pull of a whole scope, then a few `ssh` sessions), the password cache saves retyping. Your password stays in the session's locked memory and is never written to disk.

A superuser decides which tiers may use it (`tacctl console password-cache tiers`); within an allowed tier, the cache is your own choice and is off until you turn it on:

```
tacacs> console password-cache on       # saved for later sessions; starts now
tacacs> console password-cache          # your tier's permission, your setting, the limits
tacacs> console forget                  # drop the cached password now
tacacs> console password-cache off      # saved; forgets the password and stops caching
```

When your tier is not allowed, `on` is refused and the message names the command a superuser runs to allow it.

With the cache on and still empty, `ssh` and `device ssh` to a device with a pinned host key ask for your password themselves, once, before they connect:

```
tacacs> ssh sw1
Password for alice (device login):
```

The password is kept, so the next `ssh` to any device asks nothing, and the shell says `password cached for this session (console password-cache)`. Ctrl-C at the prompt ends the line with status 130 and starts nothing; an empty answer stores nothing. A wrong password stays only until the first ssh that ends with status 255, which forgets it. Without a terminal, for a device with no pinned host key, or with an option of ssh's own after `--`, ssh asks for the password itself, as it always did, and nothing is cached.

---

2. Your password

```
tacctl passwd          # asks for your current password, then the new one twice
```

One tacctl password covers every device and host in your scopes, over TACACS+ and RADIUS. Any user at any tier can change their own. The new password must differ from the current one.

Forgotten it? A superuser resets it with `tacctl user passwd <you>`. If you would rather they never see your new password, create a bcrypt hash yourself (`tacctl hash commands` prints the recipes for Linux, macOS and Windows) and send them only the hash.

2.1 A required password change

A new account, and a password a superuser has reset, usually come with a **required password change**. Until you make it:

- no device accepts your login, and no enrolled Linux host other than the tacctl server lets you in;
- on the tacctl server, the console opens with the notice `A password change is required of your account.` and starts the password change dialog straight away.

Type your current password (the one you were given), then the new one twice. Once the change is made, the session goes on as any other, and the devices accept the new password. Hosts other than the server let you in again after their next sync, which a superuser runs.

Until then, every other line is refused with `A password change is required: nothing runs here until it is made but 'passwd' (and 'exit').` Tab and `?` offer only the change. A wrong current password or a rejected new one leaves that locked prompt: type `passwd` to try again, or `exit`. Ctrl-C in the dialog ends the session.

Superusers are never held to a required password change.

---

3. What your tier allows

Your tacctl **group** sets your **tier**. Tiers, lowest to highest:

Tier	Typical person	Can do (in short)
Read-only	Monitoring, help desk	Lists and shows of users, groups and devices, `scope list`, `ssh` to devices in your scopes, `status`, `passwd`
Operator	NOC, tier 1–2	Read-only plus logs, `config validate`, `device check`, `scan`, `discover`, drift views, `backup list`
Engineer	Change engineers	Operator plus device registry and device configuration work **in your own scopes**
Superuser	Platform owners	Everything: users, groups, scopes, hosts, lockdowns, break-glass, restores

In the console and the shell, Tab, `?` and `help` show only what your tier can run. *Concepts* explains groups, scopes and tiers in full.

---

4. First commands

```
tacctl status                # is the server healthy?
tacctl device list           # the devices of your scopes
tacctl device show sw1       # one device in full: scope, state, last seen
tacctl ssh sw1               # log in to it, as you
tacctl device notice list    # open notices, each with the command that fixes it
```

Operators can add the first-line checks:

```
tacctl log failures            # authentication failures in the last 24 hours
tacctl config validate         # configuration problems; each warning names its fix
```

---

5. Conventions every command follows

Pattern	Meaning
A command family with no arguments	Prints its usage (`tacctl device`, `tacctl user`)
`add` / `remove` with `a,b,c`	Comma lists are accepted. Every item is checked first; one bad item cancels the whole command.
`clear`, `remove --all`	Always ask `[y/N]` and say what the result will be
A setter with no value	Prints the current value (`tacctl device location sw1`)
`--dry-run`, `--check`	Show what would change and write nothing. Use them before every risky command.
`--yes`	Answers the confirmation of a command that asks one, for a script
A word that is none of a command's sub-commands	One error line, a `Did you mean` hint when something fits and a `Run 'tacctl <command> -h' for usage.` line; exit status 1, no usage page (below)
Names	Lower case: letters, digits, `_`, `-`; device and host names may also use `.`
Secrets	Let the command prompt for them, or use `--hash -` / `--stdin`. Never type a secret on the command line.

A mistyped sub-command such as `device config seed jr1` (the verb is `device config shapes seed`) gets that short answer:

```
[ERROR] Unknown subcommand: 'seed'
Did you mean: tacctl device config shapes seed?
Run 'tacctl device config -h' for usage.
```

Below the first word it is answered before `sudo` asks for a password, in the plain command, in `tacctl shell` and at the console. A word that follows a name (`scope secret lab shwo`: `lab` is a scope) is left to `sudo` and the command, which ask for the password first and then answer the same way. In the shell and the console the hint names only commands your tier can run. Nothing goes to standard output. No word, `help` and a word that starts with `-` still print the usage of a command that has them.

Every change takes a snapshot first and writes an audit line, so a mistake can be rolled back (*Backups and Undo*).

---

6. Where to go next

You are...	Read
NOC, help desk, support	*Operator Field Guide*
Operator or engineer working from the console	*Day-to-Day Device Operations*
Engineer	*Engineer Guide: Device Changes*
Superuser	*Superuser Daily Operations*

Keep the *Quick Reference* open while you work.

See also: Concepts · Connecting to Devices · Glossary · `man tacctl`